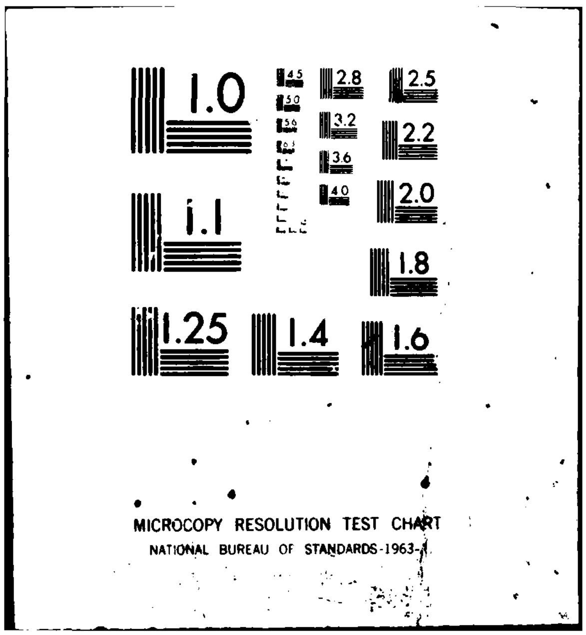
AD-A082 618 BOLT BERANEK AND NEWMAN INC CAMBRIDGE MA F/G 17/2.1
COMBINED QUARTERLY TECHNICAL REPORT NUMBER 16. SATNET DEVELOPME--ETC(U)
FEB 80 R D BRESSLER MDA903-76-C-0252
UNCLASSIFIED BBN-4342 NL

1 of 1
AD
A082618

END
DATE
FILMED
5-80
DTIC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Bolt Beranek and Newman Inc.

LEVEL II

Report No. 4342

ADA082618

Combined Quarterly Technical Report No. 16

SATNET Development and Operation
Pluribus Satellite IMP Development
Remote Site Maintenance
Internet Development
Mobile Access Terminal Network

February 1980

Prepared for:
Defense Advanced Research Projects Agency

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 4342	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): COMBINED QUARTERLY TECHNICAL REPORT NO. 16 B043536L	5. TYPE OF REPORT & PERIOD COVERED: 11/1/79 to 1/31/80
	6. PERFORMING ORG. REPORT NUMBER: 4342
7. AUTHOR(s): R.D. Bressler	8. CONTRACT OR GRANT NUMBER(s): MDA903-76-C-0252 N00039078-C-0405 N00039-79-C-0386
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Bolt Beranek and Newman Inc. 50 Moulton Street, Cambridge, MA 02138	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: ARPA Order Nos. 3214 and 3175.17
11. CONTROLLING OFFICE NAME AND ADDRESS: Defense Advanced Research Projects Agency 1400 Wilson Blvd., Arlington, VA 22209	12. REPORT DATE: February 1980
	13. NUMBER OF PAGES: 56
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): DSSW Rm. 1D 245, The Pentagon Washington, DC 20310 NAVELEX Washington, DC 20360	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE/DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer networks, packets, packet broadcast, satellite communication, gateways, Transmission Control Program, UNIX, Pluribus Satellite IMP, Remote Site Module, Remote Site Maintenance, shipboard communications

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This Quarterly Technical Report describes work on the development of and experimentation with packet broadcast by satellite; on development of Pluribus Satellite IMPs; on a study of the technology of Remote Site Maintenance; on the development of Internetwork monitoring; and on shipboard satellite communications.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

BBN-

Report No. 4342 Bolt Beranek and Newman Inc.

6 COMBINED QUARTERLY TECHNICAL REPORT ~~NO.~~ Number 16.

SATNET DEVELOPMENT AND OPERATION,
PLURIBUS SATELLITE IMP DEVELOPMENT,
REMOTE SITE MAINTENANCE,
INTERNET DEVELOPMENT,
MOBILE ACCESS TERMINAL NETWORK.

9 Rept. for 1 Nov 79 - 31 Jan 80.

11 February 1980 12 60

10 R.D. Bressler

This research was supported by the Defense Advanced Research Projects Agency under the following contracts:

15 MDA903-76-C-0252, ARPA Order No. 3214
N00039-78-C-0405, ARPA Order No. 3175.17
N00039-79-C-0386

Submitted to:

Director
Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, VA 22209

Attention: Program Management

060100 VB

1. INTRODUCTION

This Quarterly Technical Report is the current edition in a series of reports which describe the work being performed at BBN in fulfillment of several ARPA work statements. This QTR covers work on several ARPA-sponsored projects including 1) development and operation of the SATNET satellite network; 2) development of the Pluribus Satellite IMP; 3) Remote Site Maintenance activities; 4) internetwork monitoring; and 5) development of the Mobile Access Terminal Network. This work is supported under contracts MDA903-76-C-0252, N00039-78-C-0405, and N00039-79-C-0386 and is described in this single Quarterly Technical Report with the permission of the Defense Advanced Research Projects Agency. Some of this work is a continuation of efforts previously reported on under contracts DAHC15-69-C-0179, F08606-73-C-0027, F08606-75-C-0032, and MDA903-76-C-0213.

2. SATNET DEVELOPMENT AND OPERATION

A major part of our participation in the Atlantic Packet Satellite Experiment during the past quarter has been to provide support for the live demonstration of SATNET at the National Telecommunications Conference on November 31, 1979. The effort was divided into several parts, reflecting different BBN responsibilities.

Inasmuch as Clarksburg had a key role in the demonstration, preliminary tests with BBN and Comsat were conducted to debug the software and to gain operational experience with the new hardware at the Clarksburg Satellite IMP. With the present Satellite IMP software, packet headers and all control packets must be sent at the lower channel rates to permit Clarksburg to maintain reservation synchronization required by full participating members of SATNET. (Currently only 16 Kbps has been tested; further tests will be made to determine whether Clarksburg can receive reliably at 32 Kbps so as to make more efficient use of the satellite channel.) The data part of packets not destined for Clarksburg is sent at the high channel rate (64 Kbps), however, through use of the mixed-rate packet transmission feature of the PSP terminal.

Whenever Clarksburg joins the network, Tanum must be crosspatched, since it cannot receive the lower channel rate

transmissions without a PSP terminal. Furthermore, the channel bandwidth allocated for the ARPANET trunking circuit between the SDAC IMP and the London TIP must be reduced from 9.6 Kbps full duplex to 6.4 Kbps full duplex in order to leave enough bandwidth for other SATNET traffic including control traffic and monitoring traffic. When Clarksburg was removed from the network, we routinely switched to single rate 64 Kbps transmissions to provide the London TIP with better service. Daily changeovers between Tanum on the network and Clarksburg on the network were taking place, since the operational hours for Clarksburg was weekdays between 0900 EST and 1630 EST. Just prior to the demonstration, however, special arrangements were made by Comsat to keep Clarksburg on the air on a 24-hour basis, which meant that the London TIP received degraded service for the entire period. Large-scale users at London were asked to keep off the network at this time to reduce the channel load.

Hardware and software difficulties with the PTIP in November caused severe congestion in the BBN gateway and the BBNE TENEX Host, hampering the monitoring of SATNET and the testing and debugging of the demonstration software at Comsat. Since BBNE was used for loading and modifying Satellite IMPs, we disconnected the PTIP from BBNE as a precautionary measure during the final testing stages and the demonstration itself. As another precaution, we switched SATNET monitoring from the BBNE

Host to the ISIE TOPS20 Host; however, three hours before demonstration, ISIE died with no prospects of bringing it up in time. We were able, though, to transfer the monitoring onto the ISID TOPS20 Host with minimal downtime. The net result of these efforts was to eliminate congestion in the BBN gateway and the BBNE Host during the demonstration. PTIP difficulties have subsequently been fixed to eliminate the problem.

A matter of 15 minutes before the speech demonstration, the Goonhilly earth station switched transmitters. Accompanying the switchover was a warmup period in which transmission parameters drifted enough to cause the Clarksburg Satellite IMP to miss packets and thus to lose reservation synchronization with the rest of the network. At this time, we crosspatched Clarksburg out of the network to keep it from interfering with Etam and Goonhilly Satellite IMPs. After the speech demonstration, the Goonhilly transmitter had settled down, and we uncrosspatched Clarksburg to permit datagram transmission through the site to ARPANET over the SATNET channel. From then on, the demonstration worked as planned. We had BBN personnel standing by key equipment ready to perform fault diagnosis and rapid recovery procedures should that be necessary. Fortunately, the equipment for which BBN is responsible did not fail.

In addition to supporting the NTC demonstration, other accomplishments in the last quarter are described below.

With the installation of the PSP terminals at Etam and Goonhilly and with the temporary connection of the Tanum PSP terminal at Clarksburg, the Satellite IMPs have access to a wider range of information for fault diagnosis of the satellite channel. In particular, Testing and Monitoring (T&M) information is appended to each packet by the PSP terminal for characterizing packet reception parameters, including frequency offsets, AGC values, and signal and noise power estimates. Currently, each Satellite IMP keeps separate the parameters of packets originating from different Satellite IMPs, but the latest received parameters will overwrite parameters received earlier. Periodically T&M data reports are sent by each Satellite IMP to RECORDER for subsequent printout by MONITOR.

Since the default state of the PSP terminal is that T&M data are discarded rather than appended to the data packets, the Satellite IMP must enable the T&M function through the PSP terminal command module whenever T&M data are desired. After enabling the function at Clarksburg, we verified that MONITOR printout of the T&M information appended to packets at Clarksburg agrees with data measured simultaneously by Comsat personnel. We also enabled the function at Etam to examine its data, but until the PSP terminal is calibrated, the data cannot be interpreted correctly. We further tried to enable the function at Goonhilly, but without success. Since the commands were echoed back

correctly to the Satellite IMP by the PSP terminal, we believe the Goonhilly PSP terminal has been left in a state which prevents the appending of the T&M data; confirmation requires on-site attention by Comsat.

Testing with Clarksburg on the network revealed a fundamental demodulator capture effect at the lower channel rates due to the error correction capability of the Rate 1/2 Viterbi decoder. When two or more stations are broadcasting at the same time, interference among the transmissions is such that no station can be received at the high channel rate, but on frequent occasions one of the stations can be received at the low channel rate. The difficulty arises when due to local noise effects and to different receiver antenna look angles, one station does not receive the same number of request packets as the others on the network. In such a situation, that station will declare itself out of reservation synchronization sufficiently often to effectively remove itself from the network. The short-term solution to this problem is to run FPODA rather than CPODA when Clarksburg is on the network to avoid contention packets altogether. A long-term solution may require the Satellite IMP to make decisions on packet acceptability based on the T&M information appended to each packet.

We have begun conducting a series of experiments using EXPAK to determine the Satellite IMP delay performance for speech

traffic and for the traffic on the ARPANET trunking line between the SDAC IMP and the London TIP. These experiments are preliminary to developing an operational capability to measure SATNET performance on a routine basis.

During the last quarter, several hardware problems appeared, which required our attention to diagnose and, when related to the Honeywell 316, to fix. In January, the circuit between the Tanum Satellite IMP and the NDRE gateway failed; the phone company in Norway eventually discovered a cable cut in the line from Kjeller to Tanum. The 15-volt power supply in the Honeywell 316 A-drawer at Goonhilly drifted below the nominal 15 volt rating causing several crashes. On another occasion, the Goonhilly Satellite IMP crashed requiring on-site attention by BBN field maintenance personnel. It was discovered that the memory control board in the mainframe, when touched, would crash the Satellite IMP; the board was replaced, but we believe that the main cause of the problem was corroded contacts on the board. In January, the 50 Kbps Alexandria to Etam circuit developed a packet error rate at the Etam Satellite IMP as high as 2 percent; the problem was traced to the 6 volt power supply in the Honeywell 316 B-drawer at Etam, which had drifted to a low of 5.3 volts.

3. PLURIBUS SATELLITE IMP DEVELOPMENT

A significant development during the past quarter was our understanding and documentation of a delay variance problem inherent in the use of arbitrary stream intervals in PODA. This problem and candidate solutions are discussed in detail in section 3.1 below.

The design implementation was partially completed for the new host access protocol during the past quarter, and the associated coding begun. The remaining implementation is awaiting resolution of several user-oriented issues still under consideration by the other wideband network contractors, which issues are expected to be resolved shortly. We hosted a meeting on this subject at BBN on January 27, 1980; a summary of the key technical issues is given in section 3.2 below.

We attended the ESI CDR held at Linkabit Corp. in early January, and have been working with Linkabit to finalize details of PSAT-ESI interactions. More generally, we have begun the task of revising and/or adding PSAT software to accommodate the PSAT-ESI formats and ESI timing and other features. This work has been somewhat slowed by the discovery and resolution of two difficult multiprocessing software bugs. Also, as a result of discussion at the ESI CDR, we have generalized the "ID filter" function performed by the ESI for the PSAT; details are given in section 3.3 below.

We met with DARPA and Linkabit to discuss selection of the appropriate mixed-control scheme from among the candidates previously defined. While tentatively deciding to use distributively scheduled streams to solve the mixed-control problem in the near-term, subsequent work has indicated that centrally scheduled streams will also be easily achieved in the current implementation, and possibly centrally scheduled datagrams as well. A decision to include the datagram component will be deferred until a later date. (This area is being impacted by the rapid-change stream function associated with the speech prediction investigation now being carried out by Lincoln Laboratories.)

Work progressed on the low-cost PSAT study. Candidates being considered include an MBB emulating a single-processor Pluribus, a C-machine MBB, and the Butterfly machine. While the primary goal is to achieve the present full PSAT functional capability in a significantly lower-cost configuration, several other important factors are also involved. In particular, some of the configuration choices require a different software language than the one presently used in the Pluribus PSAT, offering the opportunity to achieve protocol software which is much more easily maintained and able to be adapted to different network contexts and/or data rates. Work so far indicates the most desirable approach is to rewrite the PSAT (Pluribus)

application program in C-language such that it could be run both on the MBB and on the Butterfly multiprocessor machine now under development for ARPA. We are continuing to evaluate this and other alternatives.

Other activities during this quarter included the generation of suggested revisions and additions to a PSAT/ESI interface document outline written by COMSAT, attending a wideband meeting at ARPA during January, and attending meetings with Lincoln Laboratories concerning host access issues and possible PSAT applications for DCA. We also issued W-Notes 8 and 9 on PSAT message generation (section 3.4 below) and the stream interval problem.

3.1 Stream Interval Size

There are two basic types of traffic in the Wideband Satellite Network - datagram traffic and stream traffic. Stream traffic is characterized by an interval size which determines how frequently stream messages are allocated channel time for transmission. The choice of stream interval sizes is shown below to have a significant performance impact on stream traffic in the form of a variable amount of extra delay. The present section discusses this delay dispersion problem and proposes some solutions.

The Delay Dispersion Problem

Each stream in the Wideband Satellite Network has an associated parameter called the interval size. The interval size represents the desired frequency of allocated channel time for stream message transmission. The interval is supplied as a parameter at stream creation time and is based on 10-microsecond units. For simplicity, however, all interval sizes discussed in this note will be expressed in terms of PODA frame sizes. The interval size allows the matching of allocated channel time frequency to the expected arrivals of stream messages for transmission. That is, if a stream will on the average generate 1 message every 2 PODA frames, the interval size would be 2 PODA frames. It is the existence of several stream intervals which brings about the delay dispersion problem.

The delay dispersion problem refers to the extra delay experienced by stream traffic in cases where several different stream intervals are to be scheduled for transmission at the same time. The following example will help to illustrate this.

The table in Figure 1 describes some hypothetical stream traffic and provides a scheduling timeline for those streams. The interval sizes in this example are 2, 5, 6, 10, and 15. Several streams have identical interval sizes. For example, there are 4 streams of interval size 2 and 10 streams with an

INTERVAL SIZE (in frames)	# STREAMS PER INTERVAL	FRAME # SCHEDULING										
		30	31	32	33	34	35	36	37	38	39	40
2	4	X		X		X		X		X		X
5	4	X					X					
6	4	X						X				
10	10	X										X
15	10	X										

Figure 1

interval size of 15. The scheduling timeline is drawn as a box diagram. An "X" indicates that a stream is to be scheduled in a particular frame.

For simplicity, several assumptions have been made in this example. First, it is assumed that all intervals have been initialized to start at frame 0; thus the scheduling "collision" at frame 30. Second, integral interval sizes have been used although this is not a necessary constraint. Third, all stream message slots have the same size. Finally, there is sufficient channel time for six stream message slots per frame.

Streams are scheduled according to transmission times based on the interval size. Streams with identical transmission times are kept in First-In First-Out order. Using this FIFO scheduling, the channel timeline depicting stream scheduling appears in Figure 2. Numbers below the frame timeline are frame numbers. Numbers appearing in the frames depict the interval scheduled. Numbers appearing above the braces indicate the frame number that stream messages were to be scheduled for. The average and maximum extra delays are tallied in Figure 3. Note that some of the intervals experience relatively large extra delays. This delay can significantly degrade system performance.

Two things should be pointed out before continuing. First, the "collision point" of intervals can be determined by the least

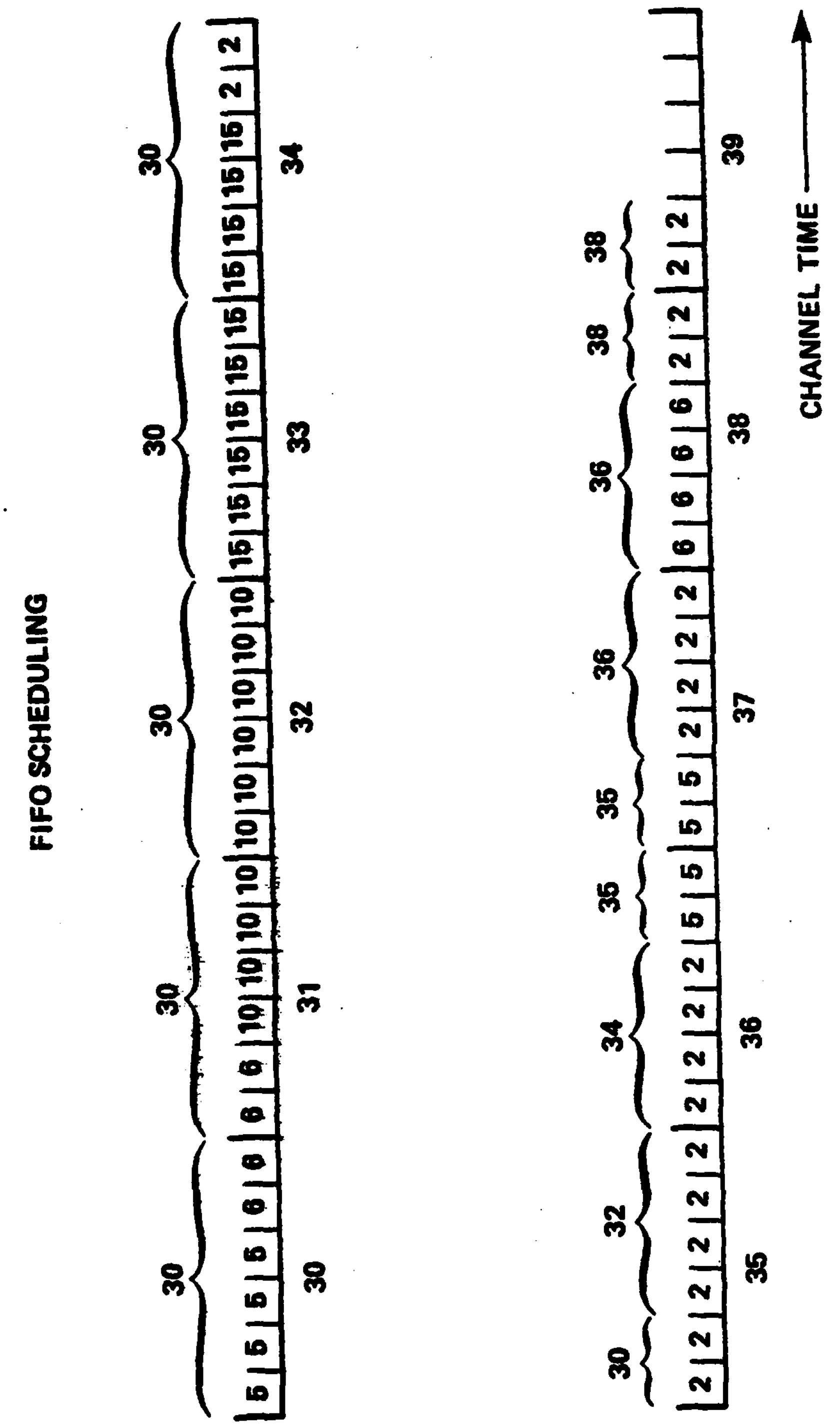


Figure 2

EXTRA DELAY FOR FIFO SCHEDULING

INTERVAL SIZE (in frames)	AVERAGE EXTRA DELAY (in frames)	MAXIMUM EXTRA DELAY (in frames)
2	2.2	5
5	.75	2
6	1.25	2
10	1.6	2
15	3.4	4

Figure 3

common multiple relationship. For example, the first major "collision" occurs at frame 30 (LCM for 2,5,6,10,15). Second, even if the interval starting points are cleverly staggered, or interval sizes are relatively prime to each other, the delay effects can still be significant.

Three approaches to this problem are now discussed. The first approach assumes a need to support arbitrary interval sizes and uses a scheduling strategy to keep delays proportional to interval sizes. The second approach restricts the choice of interval sizes to powers of 2; allowing only this subset of arbitrary interval sizes, the delay dispersion problem can be eliminated. Finally, the use of a single system interval is discussed. This last approach not only eliminates the delay dispersion problem, but also significantly simplifies the PODA scheduling function in the PSAT.

Problem Solutions

A. APPROACH 1 - ARBITRARY INTERVAL SIZES

If the flexibility of arbitrary stream intervals is necessary, the delay dispersion problem can be minimized, not eliminated. The amount of extra delay is determined primarily by the stream interval scheduling policy. While several strategies exist, the approach explored here is the shortest-interval-first scheduling

policy. This approach ensures a delay proportional to the interval size. Smaller intervals see short delays while larger intervals see increasingly longer delays. Using the same example as in Figure 1, the scheduling timeline for this approach is shown in Figure 4 and the associated delay figures appear in Figure 5.

This approach allows the full generality of stream interval sizes at the expense of some performance degradation due to delay. Of the three approaches it is computationally the most complex for the PSAT.

B. APPROACH 2 - POWERS OF TWO INTERVAL SIZES

By restricting the choice of interval sizes to powers of 2, the delay dispersion problem is eliminated. That is, the interval size = 2 to the power p where p is some non-negative integer. If intervals larger than 16 frames are unlikely, there are basically 5 possible interval choices: 1, 2, 4, 8 and 16. This approach can most easily be illustrated by using a box diagram. The example in Figure 1 does not conform to the powers of 2 constraint so other interval sizes will be used. However, the four initial assumptions specified above still hold. Stream intervals and identification are tabulated in Figure 6. Since interval sizes are no

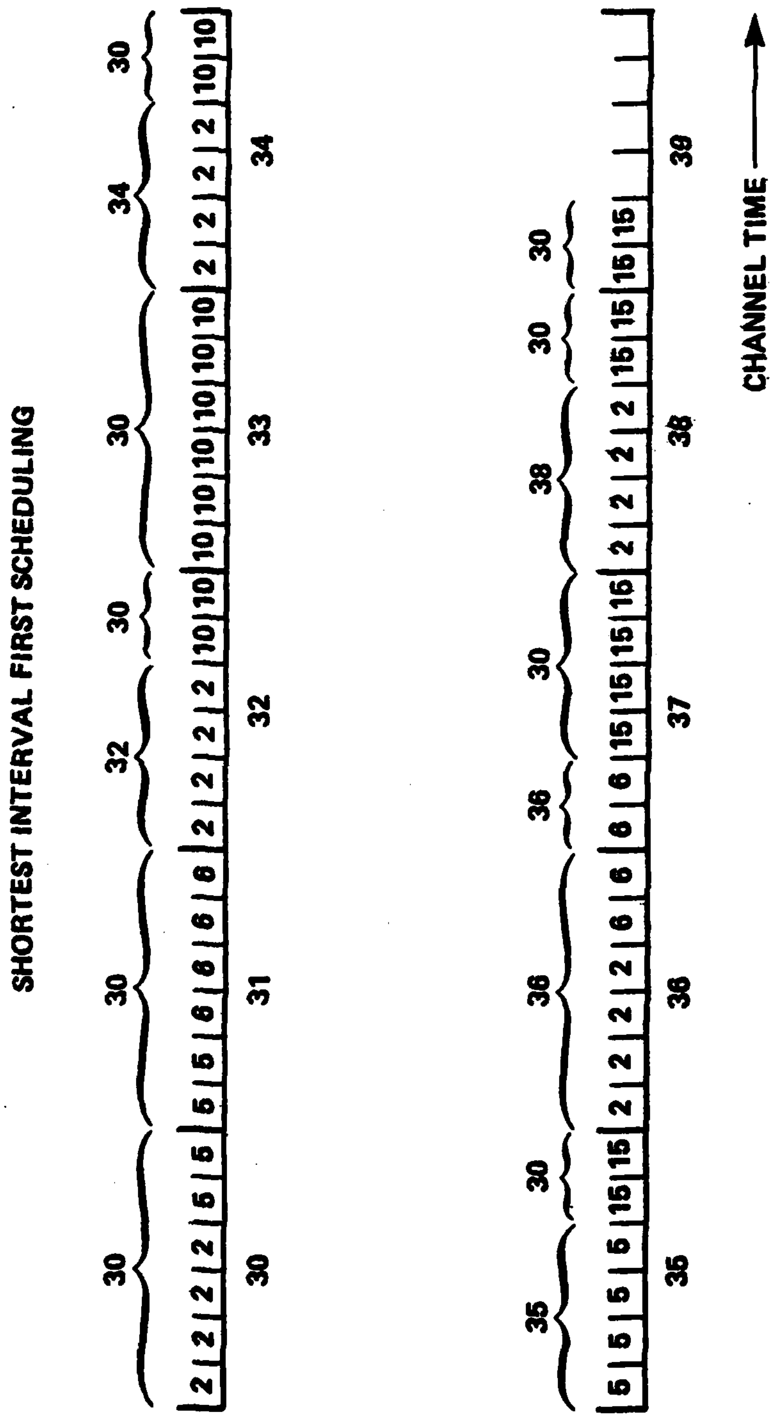


Figure 4

EXTRA DELAY FOR
SHORTEST INTERVAL FIRST SCHEDULING

INTERVAL SIZE (in frames)	AVERAGE EXTRA DELAY (in frames)	MAXIMUM EXTRA DELAY (in frames)
2	0	0
5	.25	1
6	.75	1
10	3	4
15	7.2	9

Figure 5

POWERS OF TWO INTERVAL SIZES

STREAM ID	INTERVAL SIZE
A	1
B	2
C	2
D	2
E	4
F	8
G	4
H	1
I	2

Figure 6

larger than 16, one can visualize the scheduling timeline repeating after every 16 frames. The timeline for this approach is shown in Figure 7. The letter contained in each box identifies the stream scheduled for a particular slot.

It is apparent that no stream experiences extra delay in this approach. However, the flexibility of arbitrary stream interval sizes no longer exists. The computational complexity of this approach for the PSAT lies somewhere between approach 1 and approach 3.

C. APPROACH 3 - SINGLE SYSTEM INTERVAL SIZE

At the opposite extreme of allowing arbitrary stream interval sizes is the use of a single system interval size. Here, each stream has the same interval size. This system interval must be small enough to accommodate the smallest interval desired by hosts to minimize the impacts of lost packets.

POWERS OF TWO INTERVAL SCHEDULING

SLOT #																	
5																	
4	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
3	G	I		I	G	I		I	G	I		I	G	I		I	G
2	D	E	D	F	D	E	D		D	E	D	F	D	E	D		D
1	B	C	B	C	B	C	B	C	B	C	B	C	B	C	B	C	B
0	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

FRAME #

Figure 7

For stream source packets generated with larger or smaller intervals, the host would perform a translation of the stream interval by either segmenting or aggregating the stream packets such that they match the system interval. Since this approach is simple with respect to channel scheduling, no illustration is needed. Computationally, this is the least complex of the three approaches for the PSAT.

Discussion

Two fundamental questions concerning the interval size choice are: 1) Is the generality of arbitrary stream interval sizes necessary or is a subset of these choices sufficient for the user community? 2) Is a single system stream interval too restrictive?

These questions were raised and debated at a recent Wideband Network Host Access Meeting. Certain conclusions are apparent from this discussion. The users did not see a real need for the flexibility of arbitrary stream interval sizes. The consenus seemed to be that only a few stream interval sizes would be used. However, no real disagreement was voiced with the idea of a single system stream interval. The apparent problem with the single interval seemed to be streams requiring larger or smaller intervals. The concept of aggregation/segmentation of messages by hosts seemed to alleviate this problem, however.

The choice of stream interval size(s) is highly dependent on use requirements. As voice traffic is expected to be the primary source of stream traffic, the choice of stream interval size(s) should be influenced by the characteristics of speech. Users at the meeting indicated that the current choices of PODA frame size (either 20 or 40 milliseconds) would be a good range for interval sizes. In fact, a 20 msec interval size is considered "ideal" while a 40 msec interval size would be satisfactory. The characteristics of other types of stream traffic also require examination. Unfortunately, information regarding interval sizes is presently unavailable for video and other potential applications.

As indicated above, arbitrary stream intervals are apparently not necessary. The decision then becomes a choice between the powers of 2 approach and the single system interval. The powers of 2 approach is somewhat appealing because of its flexibility in providing several interval sizes should they prove neccessary for future applications of streams. In addition, the processing bandwidth for scheduling is not too large. Whether even this flexibility is necessary relative to the simple single interval choice is uncertain. Thus, the decision as to the best approach should await further understanding of stream use requirements.

3.2 Host Access Protocol

This section summarizes recent developments or revisions to the host access protocol (HAP). Protocol level numbers correspond to X-25 definitions.

1. Error Control

<u>Type-of-Service (Satellite Channel):</u> Users will be able to select from several different reliability levels (bit error rates), with actual values and bandwidth penalties determined later. A separate bit will be used to designate whether messages with detected data errors should be delivered by the network to the destination host, and will apply to all reliability levels. (A still unresolved question is whether messages with detected errors should be marked as such in their headers.)

<u>Host-PSAT:</u> For local connections, no level-2 error control will be used. For remote connections or those with significant error rates, HDLC is tentatively defined for use at level 2 (subect to protocol chip development for megabit rates). For either type of connection, a 16-bit two's-complement checksum will be used at level 3 to detect errors in all Host-PSAT control information (message headers and all words of

separate control messages); messages with bad level-3 checksums will be discarded but not acknowledged or retransmitted.

2. Multilinking

Multiplexing/demultiplexing of a logical flow across two or more physical links will be done at the level-3/level-4 interface. That is, level-3 functions such as flow control, monitoring, and restarts will be done separately for each physical link.

3. Port Expansion

The PSAT will support a direct 'local net' connection of multiple hosts through a single PSAT physical port, with a separate instance of HAP executed by the PSAT for each host. A level-2 header will be used to carry local addressing information. For compatibility with available hardware chips which would be used for address filtering, the local destination address field will consist of the first eight bits of the level-2 header. (The following 32-bit header is under consideration: first byte is the local destination address; second byte is the local sender's address; third and fourth bytes are reserved for future use.)

4. Flow Control

A non-blocking selective acceptance/refusal (A/R) flow control scheme will be provided, using mod 256 message numbering (zero excluded). Block acceptances and refusals (of the same refusal type) will be used when appropriate to minimize the exchange of explicit control messages. A host may inhibit A/R signaling for specific messages by using a value of zero in their message number field, and may tell the PSAT to use zero for all messages sent to the host. Additional flow control information, such as an indication of the acceptable message rate for a particular destination, may be added later.

Certain kinds of refusals and stream phasing messages will need to be returned by the PSAT even if message number zero is used; details for this are still to be worked out.

5. Monitoring

A monitoring scheme will be used which allows both the host and PSAT to determine error rates and lost messages in both directions, independently of whether A/R signaling is used. Each side maintains a cumulative 16-bit counter which tracks the number of distinct

messages (control and data) sent, and cumulative counters which track the number of messages received correctly and kept, received correctly but discarded, received with a bad level-3 checksum, and (if applicable) received with a bad level-2 checksum. The counter valves are exchanged in periodically sent status messages such that synchronism of send/receive counters is maintained, allowing subsequent determination of error rates and dropped messages.

6. Maximum Message Size

It has been proposed that the maximum message size accepted by the network be defined as (400 + 2**N) for N either 13 or 14 -- e.g., either 8,592 or 16,784 [choice of N to be determined later]. The 400 bits is intended to provide room for all local, internet, and higher-level headers while allowing actual data to be a power of 2.

7. Inter-Arrival Time Control

The possibility of providing a mechanism which allows delays to be introduced between consecutive arrivals at the I/O level is under consideration. The purpose of this is to slow down the arrival rate to that which can actually be handled by the receiver. One

method is for each receiver to start and stop the clock used by the sender. Another possibility is the use of a 'pause' parameter, which would be applied at the software level as is now done in SATNET.

3.3 ESI Message Filtering

Due to the varying requirements and processing capacity of nodes in the wideband satellite network, it is useful to inhibit downlink traffic of no interest to a given node from reaching that node, by blocking it in the ESI. A number of possible mechanisms were examined, and the following was chosen on the basis of providing the most flexibility for the PSAT, while requiring minimum knowledge of the burst format and little processing in the ESI.

Filter Operation

The filtering mechanism is composed of two stages, the first eliminating most of the unwanted bursts, the second blocking reception of the station's own transmissions. Filtering is done on a burst basis, and is controlled by a 12-bit field in the burst header. The two high bits of this field are the filter level, the low 10 bits are the filter data field.

The first filter mechanism is controlled by the filter level field, and works as follows. An incoming burst has its filter

level field extracted and compared with the current value of the station filter level (a parameter set by PSAT command to the ESI). If the incoming burst has a higher value than the station level, the incoming burst is completely ignored, and is dropped by the ESI. If the burst level is lower than or equal to the station filter level, then the burst is passed to the second filtering test.

The second filter is controlled by the filter data field, which is matched against a value set by PSAT command to the ESI. If the field matches the value, only the burst control packet is passed by the ESI, and subsequent packets in that burst are dropped. If the field does not match the set value, the whole burst is passed on to the PSAT. This is intended primarily to keep stations from having to hear their own transmissions, in which they are presumably not interested.

Filter Use

The mechanism described above is expected to be used in the following way. PSATs will put their station ID in the filter data field, and will set the local match value (in the ESI) to also be their station ID. One bit in the field will be used to enable or disable matching, by having station IDs always be 9 bits long with an assumed high-order 10th bit of 0. Therefore, if a station wants to hear its own transmission, it need only set

the high bit in the transmitted filter data field, which will force the match operation to fail. For stations with high-bandwidth transmissions, this matching operation results in a significant reduction in processing requirements, since stations which are sending high-bandwidth traffic do not have to (uselessly) receive it.

The first stage filtering will be used to reduce incoming traffic at stations with reduced processing capabilities. At the current time, there are two classes of stations which might use this: those using centralized datagram scheduling, and those which are running the satellite loader/dumper program. The fully capable third class of stations is the distributed scheduling group. The loader only wants to receive leader bursts, loader messages, and (possibly) its own transmissions. Centralized stations also want to receive data and stream traffic, and possibly also setups. Distributed stations need to hear all traffic, including control packets. To accomplish this, leader and loader bursts have a filter level of 0, data and stream message bursts (also ranging and possibly setups) have a level of 1, and control and setup bursts would have level 2. PSATs would set their local ESI receive filter level according to what they were running: 0 if the satellite loader/dumper, 1 if centralized scheduling, and 2 if distributed scheduling.

3.4 PSAT Message Generators

The PSAT currently provides two types of traffic emulation processes, for use in system measurement and experimentation. The MSGEN process acts as a message source/sink for both stream and datagram traffic, whereas the SPEECH process generates only stream traffic, but provides an additional layer of control structure to maintain the state of the stream. This section discusses the common characteristics and user-specified parameters of each process, for purposes of comparison.

The two traffic emulation processes share a number of characteristics, including periodic activation, parameter control and statistics collection. Both processes are issued WAKEUPs by a system timer strip (the Pluribus equivalent of an interrupt routine) once every 25.6 milliseconds. They then check to see whether it is time to generate data, and check their Host Output Queues for data returned by the system. Various parameters and run-time options are defined with the PARAMETER macro, which enables the Measurement internal host and the EXPAK Tenex/TOPS-20 process to set up initial algorithm parameters and vary them while a generator is actually running. Finally, cumulative statistics (CUMSTATS) are periodically gathered by the Measurement internal host and delivered to the EXPAK process for later analysis and data reduction.

The network access interface for all PSAT internal hosts, including the message generators, allows stream and datagram message transmission using an internal protocol similar to the draft Wideband Network Host Access Protocol, but differs from the protocol defined in such areas as flow control and link monitoring. No ACCEPTANCE or REFUSAL messages are used for datagram messages, and refused datagram messages are simply discarded at the source. Stream messages which would have been refused due to message lengths greater than the current stream allocation are currently truncated by the PSAT Host Protocol Modules (HPM) rather than discarded, to allow experimentation with dynamic stream allocation techniques. Link monitoring messages are eliminated, as the PSAT currently passes buffer pointers through the message processing software, and hardware errors which would corrupt this shared-memory link are better detected by the Pluribus run-time reliability system. With these exceptions, the HPM deals with internally generated traffic the same way it does traffic from external sources.

More than one of each process may be created at system initialization in the PSAT, with different parameters (including network addresses) specified for the different internal hosts. An important issue which is currently under investigation is the amount of processing bandwidth necessary to run the different message generators. It should therefore be noted that the

processes specified here are subject to change if the current implementation diverts too much processing power from the mainline tasks of message transmission and channel scheduling.

MSGEN Process

The MSGEN process is implemented using an array of parallel parameter tables to simulate a number of distinct message sources. For each source, the following parameters are specified:

- Local network source host address
- Local network destination host address
- Internetwork destination host address
- Destination network
- Internet protocol
- Type of service
- Message length or distribution of lengths
- Transmission interval or distribution
- Data pattern (16 bits) to fill in data portion of message

The message lengths presently available are uniformly distributed or constant, although a geometric distribution may be added later. Minimum and maximum message lengths are as specified in W-Note 4. Similarly, uniformly distributed or constant transmission intervals are provided, with a lower bound of every 25.6 milliseconds imposed by the current process control strategy. It should be noted, however, that any arbitrary series of transmission intervals and message lengths may be generated by the aggregate of individual generators, each with different length and interval parameters. Currently, no end-to-end statistics are implemented in MSGEN.

The MSGEN process treats datagram and stream messages identically, with the only difference being the message code in the type of service leader word. MSGEN assumes that any streams used have been set up previously, and are maintained manually by the experimenter using EXPAK in conjunction with the Stream Service Host, which is another PSAT internal host.

SPEECH Process

In contrast to MSGEN, the SPEECH process handles stream messages exclusively, and incorporates an internal finite state machine which interacts with the PSAT Service Host to create, change, and delete streams. The EXPAK control interface consists of a process control word in which various flags are set by the EXPAK user to start the stream set-up, restart the process, issue stream change messages, initiate stream teardown, and control run-time options.

The process implements a talker activity model and speech predictor process which have been developed at Lincoln Labs and are described in [1]. The SPEECH process has been developed to aid in the advanced systems experiments and, in particular, to investigate the efficiency by which small numbers of full-duplex speech users may be supported on the Wideband Network, by using speech traffic emulation software to drive a dynamically allocated stream. More detail on the structure of the PSAT

speech generator will be provided along with the results of the speech-oriented experiments.

The relevant parameters in the speech generator are:

- Probability density function of talkspurt and silence durations
- Total number of talkers
- Simulated vocoding rate

Given a sampling rate of 25.6 milliseconds, the vocoding rate is divided by 1/.025, or 40, to determine the simulated speech parcel size for each active talker in a sampling frame. For example, with a vocoding rate of 16,000 bps, each active talker will generate a 400 bit (25 word) parcel every sampling frame. Stream data message parameters include only local network source and destination host addresses, as the internet stream protocol is still under development.

The stream created by the SPEECH host is controlled by the following parameters, which may be set up initially for stream creation, and subsequently changed via stream change messages:

- Stream slot size
- Stream transmission interval
- Stream priority and holding time
- Key used for access control

The SPEECH process also incorporates an activity predictor which automatically generates a stream change message when the expected traffic load differs from the current allocation by a fixed amount. The key parameters to this algorithm are the prediction bias, and the the amount of time expected for the stream change message to take effect. The statistics gathered include end-to-end delay on individual messages; total throughput; messages lost, truncated or received out of order; total lost speech; and various algorithm monitors.

REFERENCES

[1] C. J. Weinstein, "Fractional Speech Loss and Talker Activity Model for TASI and Packet Switched Speech," IEEE Trans. on Communications, Vol. COM-26, No. 8, pp. 1253-1257 (August 1978).

4. REMOTE SITE MAINTENANCE

4.1 Remote Site Module -- General

The principal activities in this period have been the installation of the CINCPACFLT system and the upgrade of the software at NOSC and at the Naval Postgraduate School.

The NOSC software had not been significantly modified during the past year. During that period, an experimental version of the UNIX operating system, which combined the standard BBN modifications with the VTUNIX system in use at NOSC, was prepared. At the time of the NPS system installation, it was decided that this approach should be abandoned for several reasons. These included the performance penalty which the virtual terminal system exacted from all running processes, which was significant. Furthermore, and the most interesting feature, the use of multiple windows on a single display was limited to the highly intelligent Genisco, and could not be used on the Ann Arbor or similar "dumb" terminals. The alternative selected, to provide a user-level screen handler to support virtual terminal functions, is described below.

In the course of preparing for the demonstration of the system at NPS, the local staff encountered some difficulties which were not completely repaired following a system crash. After reviewing the condition of the software and evaluating the

multiple inconsistencies, BBN decided to perform a system reload over the open network. This required a considerable effort, and has provided some useful insights into the problems of transferring large systems over the net. The program and documentation sources are contained on a 30 megabyte file system, which was transferred overnight to NPS at the beginning of the upgrade. The entire kernel and most of the utilities (of which there are approximately 200) were recompiled and installed on a running system. The over-the-net reboot worked satisfactorily, although it was never attempted unless someone was actually at the computer to allow a graceful fall-back.

4.2 Virtual Terminal Screen Handler

BBN has developed a screen handler that works on the DEC VT52 and VT100 as well as on the Ann Arbor terminal. It runs under the UNIX time-sharing system and allows the user to divide the screen into several rectangular regions and to treat each region as an independent output device. The user issues simple commands to attach the keyboard logically to any of the windows.

The screen handler can be modified to accept other terminal types. The only functions required of the terminal hardware are: home, up, down, left, right, and cursor position.

The user invokes the screen manager from command level. The screen manager is told the terminal and window type. It then

clears the screen and sets up a full screen window, through which the user interacts with the UNIX system.

There are two initial window types, which are selected on the command line. The more general type is the pre-login window. In this case, when a window is created, it looks like an ordinary terminal connected to the UNIX system in the pre-login state. After log-in the terminal handler uses the same mechanisms for this window as are used for network terminals. At the end of a session the user logs out, and the login message will appear again.

The other window type simply starts up a shell. In this case, the user does not log in, but interacts with the shell. The user may ask for a login message, or a new shell, at any time in any window.

The windows look to the system like small versions of the original terminal. All cursor motion sequences work (mapped into the window) and the alternate character set is available for the VT52 and VT100. This allows programs which depend on such features, such as NED, to run without change.

To simplify implementation, only the information currently displayed is stored. If a new window overlays part of the text in an old window, the text will not reappear if the new window is deleted. For similar reasons, some possibly interesting

features, like overlapping or non-rectangular windows, were not included.

Screen manager commands consist of single letters with modifiers. The assignment of characters can be easily changed. In the description below, these single characters are shown as capitalized character strings to improve readability. Most characters are directly transmitted to the window as they are typed (including the quit and interrupt keys). Commands are preceded by an ARG (ARG ARG will send a single ARG to the window). For example, ARG <number> <side> <type> creates a subwindow. <Number> specifies the number of rows or columns to break off from the current window. <Side> specifies where to place it (TOP, BOTTOM, RIGHT, LEFT). <Type> is either PTY or SHELL. (PTY is a prelogin window and SHELL is already logged in.) To make a new shell window in the rightmost 30 columns, one types ARG 30 RIGHT SHELL. The shell is a child of the shell in the original window. If one types ARG SHELL or ARG PTY, a new pty or shell is created without changing the window size.

The command ARG MOVE moves the user to the next window. ARG DELWIN will only work in a childless window that is the youngest child of its parent. ARG CLRWIN clears the window. ARG BORDER and ARG UNBORDER will add or remove a border from a window. The number of borders a window can have is limited only by the size of the window. The window can never be less than 1 line or 1

column. ARG WRAP and ARG <number> SCROLL affect the scrolling of the window. Wrapping is the initial state since it is much more efficient. ARG ERASE toggles whether or not the next line down gets erased by a linefeed; this is initially on since it makes the script more readable when wrapping. ARG AUTO toggles whether or not text that runs off the right edge of a window will automatically go to the next line. Each of these commands affects the current window only. ARG REPAINT can be used to repaint the entire screen (all windows) if by chance the contents are destroyed by static or a system broadcast message.

The screen manager uses certain features of the BBN version of the UNIX system, including the await and capacity system calls and pseudo Teletypes (ptys).

5. INTERNET DEVELOPMENT

5.1 VAN gateway

During the last quarter, we implemented routines to interface the gateway software with level 3 of X.25 software provided by UCL. The UCL software provides level 1 (hardware driver), level 2 (link control) and the following functions for level 3:

- listening for incoming connections;
- accepting or refusing incoming connections;
- initiating outgoing connections;
- sending and receiving data packets;
- sending and receiving interrupt packets;
- detecting or initiating a disconnect.

While the X.25 software is currently in use at UCL, it has yet to be tried elsewhere. We expect to interface the software to a test facility in Telenet which is an X.25 network, through use of a special debugging line as soon as the line and modem are installed.

The main gateway, which provides the dynamic routing and reporting capabilities, requires the following actions from a network interface:

- bring the network connection up (e.g. closing ready relay and exchanging NOPs for 1822);

- obtain buffers and initiate reading (receiving);

- handle read completion (packet received) by checking the local header for validity and by passing the packet to the main gateway for forwarding;

- accept packets from the main gateway for output by composing the local leader, initiating a write (send), and releasing buffers when successfully sent.

The gateway routines which interface to an X.25 network were written to implement these gateway functions as level 4, above the X.25 connection. Because X.25 is a connection oriented protocol, the capability of bringing up the network connection is implemented as initiating a single connection to a preset address. Hence, any incoming calls will be refused for the present. Once the connection is established, data packets can be sent and received over that path. The host on the other end of the connection may be another gateway or a re-distribution center for datagrams. To match the environment of the rest of the gateway, the new routines were written in BCPL language.

After we coded and compiled the X.25 network interface package for the gateway and loaded all the software together to create the gateway, most of the available memory in the LSI-11, which is 28K 16-bit words, was occupied by code. Since so little space was left for buffering, the gateway would certainly fail to provide any useful service. In an effort to gain more buffer

space, the module named "dialog", which served to type out internal tables on the local terminal when requested, has been eliminated. Much of this information is already provided through the CMCC programs, and the data can be obtained without use of the local terminal and the dialog module. When debugging has proceeded to the point of setting up a real connection, we expect that extra buffers can be assigned to the UCL software package through the elimination of superfluous debugging code.

5.2 CMCC Development

We completed the implementation of the initial phase of the Catenet Monitoring and Control Center (CMCC) software, which provides a basic information-gathering system to be used in the analysis of internetwork system operations and the diagnosis of system malfunctions. This phase of the CMCC knows about the catenet only in terms of the gateway up/down status and the gateway throughput statistics components. Among the features included are:

- display of gateway throughput and status reports
- limited status display of the catenet
- multiple user access to information
- capability of sending specific inquiries to gateways
- generation of a log file on demand.

The software has been undergoing extensive tests and is now ready to be used on an operational basis.

The CMCC software consists of two parts: a control process, communicating directly with the gateways, and a number of user terminal processes, communicating with the control process. Each terminal process sends requests to the control process to obtain information from the gateways and displays this information when it is received. In addition, the terminal processes can be used to display information stored by the control process.

Users can make requests for gateways to start or stop sending regular reports, to enable or disable the sending of event messages (traps), and to answer single inquiries. The regular reports consist of packet throughput statistics. Trap types include interface up or down and neighbor gateway up or down. The inquiry types presently available are the routing table, the interface up or down status, and the gateway descriptions. If a gateway does not implement the necessary monitoring facilities, the user will be informed that his information request of that gateway cannot be satisfied. To avoid conflicts, there are software locks to prevent more than one user at a time from manipulating reports or traps in a gateway.

Upon demand, a selective set of messages coming to the control process is recorded into a log file in the same format as the terminal display. The log file enables the user to analyze system operation and to diagnose system malfunctions _a posteriori_.

Using the report and trap information, the CMCC maintains a representation of the up/down status of all the gateways and their interfaces for subsequent display by a terminal process. Whenever a gateway or interface goes down or comes back up, a terminal alarm is generated for alerting CMCC personnel.

Performance of the current implementation has been enhanced over previous versions by several modifications during this quarter. First, we redesigned the program to reduce the number of forks used from three to one. Second, we redesigned the mechanism with which the control process communicates with the terminal processes so that the number of times the central queue is locked is considerably reduced. As a result of these modifications, the contribution to TENEX load averages when running this software is at an acceptable level.

This phase of the CMCC software has been documented in the internet reports IEN 131, "Gateway Monitoring Protocol", and IEN 132, "The CMCC Terminal Process", which were written for distribution at the February internet meeting.

6. MOBILE ACCESS TERMINAL NETWORK

6.1 Summary of Past Quarter's Work

As part of our participation in the Mobile Access Terminal Network (MATNET) during the last quarter, we placed emphasis on two items associated with the development of the Red subsystem. First, expansion of the Honeywell 316 architecture to accommodate Satellite IMP programs running in an MBB with a 64K-word address space was specified (see Section 6.2). Second, the design specification of the Red/Black interface was completed (see Section 6.3).

During the last quarter, we also finished the design of the satellite simulator and we began development of new op-codes along with new microcode for the MBB. Assembly of the satellite simulator has begun, and parts for it are continuing to arrive. The device is expected to be finished in April,

6.2 64K-word MBB Operation

The addition of new features in the Satellite IMP over its entire development period has been accompanied by the conversion of buffer space to code space to the extent that buffer space is no longer over abundant. Although a 32K-word memory space is currently sufficient for running the Satellite IMP program for satellite processing nodes within SATNET, no space is available

for adding the 1822 local host code and the code to service the Red/Black interface necessary for satellite processing nodes within MATNET. To overcome this problem, we have specified that the MBBs serving as Red processors in MATNET have 64K words of memory, even though the Honeywell 316 instruction set conforms to a 32K-word memory address space. The expansion of the Honeywell 316 architecture to permit addressing of a larger memory space within the MBB was specified during this quarter.

With 20 address lines and 20-bit memories, it is feasible for the MBB to address up to 1M words of memory directly. Increasing the memory address space of the Honeywell 316 architecture to very large sizes can be done in several different ways. One particularly attractive method is simply to expand the number of bits in the page descriptor field from 9 bits to 13 bits to increase the page size from 512 words to 8192 words. Another method is to partition memory into disjoint segments of 32K words (similar to and in addition to the partitioning of memory into pages). While each of the above is appealing with regard to removing any practical space restriction on satellite IMP programs, the effort to change all the TENEX supporting programs (assembler, loader, and debugger) is unjustifiably large for the MATNET project. Hence, we have compromised on a third method which has minimal impact on the TENEX supporting programs but restricts the memory address space to 64K words; memory

expansion beyond that amount for MATNET appears to be unnecessary.

Although the Honeywell 316 has 16 address lines and 16-bit memories, the instruction set interprets the sign bit of a word which references memory as an indirect reference bit and not as a 16th address bit. Whenever there was no ambiguity with the indirect referencing function, the programming convention in many instances has been to take advantage of this spare bit as a flag. (Before the words are used to reference memory, though, the flags in the sign bits are masked off.) A simple modification of the Honeywell 316 architecture which permits the addressing of up to 64K words is to designate the sign bit of a word which references memory solely as a 16th address bit. One consequence of this procedure is that multilevel indirect addressing is no longer acceptable, which fortunately rarely occurred within the Satellite IMP program. Conversion of the Satellite IMP code consistent with the new architecture has already been completed, requiring the removal of all cases of multilevel indirect addressing and all instances where flags were placed in the sign bits of words referencing memory. Although testing of the Satellite IMP was performed on a 32K-word Honeywell 316 to ensure that program operation was not affected, true 64K word operation must wait for delivery of an MBB with enough memory and with the microcode to reflect the new architecture.

Assembly of the Satellite IMP code required assembler modifications to recognize 64K-word MBB operation. These modifications were implemented during this quarter, concurrently with improvements to reduce the assembly computation time by 25%.

6.3 Red/Black Data Transfer Format

The data packet transfer between the Red subsystem and the Black subsystem is constrained by the security equipment, the codec, the interleaver/deinterleaver, and the processor input/output circuits. Each of these items must be taken into account for specifying the data packet format.

The packet format used in the transfer of data from the Red subsystem to the Black subsystem is shown below.

```
+--------------------+------+----------+------+-----+
|     FILL BITS      |ZEROES|          |  KG  |FILL|
|KG PREAMBLE AND PREP|      |   DATA   | FLUSH|BITS|
|      842 bits      |6 bits|          |2 bits|    |
+--------------------+------+----------+------+-----+
 [  848 bits = 106 WORDS    ] [ integral # of WORDS]
```

The packet begins with 842 fill bits, in place of which the KG-36 crypto equipment outputs the preamble and security key to the Black processor. The next six bits are zero to allow the receiver to synchronize on byte boundaries. Following the fill bits are the data to be transmitted over the satellite channel. Interleaver operation restricts the acceptable data lengths by requiring the transmission to be segmented into blocks, where the

length of a block is either 252 or 504 bits for Rate 1/2 code or Rate 1 code, respectively. Up to sixteen blocks equivalent to 8K bits of data can be accommodated in a single packet. The first interleaver block contains the compressed KG key and other overhead bits and therefore contains fewer data bits. The crypto unit inserts a two bit delay in the data path, requiring two flush bits to assure that all the data are transferred to the Black processor. Since the Red processor transfers integral numbers of sixteen bit words, up to 15 fill bits are added to the end of a packet to reestablish the transfer onto word boundaries; these bits, however, are not transmitted over the satellite channel.

In the Black processor, the crypto key is compressed to 64 bits, the information is encoded and interleaved, and the packet format is transformed to the following prior to transmission over the satellite channel.

PACKET LENGTH	KG PREAMBLE	EXTRA KG PREP	ENCRYPTED BITS	CODER FLUSH
8 bits	64 bits	2 bits		6 bits

A packet length field (in units of interleaver blocks), an extra 2 bits of KG prep, and 6 bits of coder flush are affixed to the data. With Rate 1 code, the final 6 bits of coder flush are absent.

At the receive side, the black processor deinterleaves and decodes the information, strips off the packet length field, and expands the crypto key to 841 bits. The packet format used in the transfer of data from the Black subsystem to the crypto is:

```
+------+--------+-------+-----------+-------+------+----+
| FILL | KG KEY | EXTRA | ENCRYPTED | FILL  |  KG  |FILL|
|      |        |KG PREP|   BITS    |       | FLUSH|BITS|
|7 bits|841 bits| 2 bits|           |15 bits|2 bits|    |
+------+--------+-------+-----------+-------+------+----+
 [  106 words  ] [    integral number of 8-bit bytes    ]
```

Seven bits of fill are prefixed to the packet by the Black processor to allow it to begin on byte boundaries. Fifteen bits of fill, to assure that at the end of a packet an entire sixteen-bit word is received by the Red processor, and two bits of KG flush are also affixed to the packet. The crypto strips off the key and passes the data to the Red subsystem with the following format.

```
+-------+-----------+-------+----+
|KG FILL|           | FILL  |FILL|
|       |   DATA    |       |BITS|
| 3 bits|           |15 bits|    |
+-------+-----------+-------+----+
```

The packet received by the Red processor starts with three bits of KG fill, followed by the data bits and an appended fifteen bits of fill (all zero). Additional fill bits to allow the Black processor to complete an eight bit transfer are also added but ignored by the Red processor.

DISTRIBUTION
[QTR 16]

ARPA
Director (3 copies)
Defense Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209
Attn: Program Manager

R. Kahn
V. Cerf
J. Dietzler

DEFENSE DOCUMENTATION CENTER (12 copies)
Cameron Station
Alexandria, VA 22314

DEFENSE COMMUNICATIONS ENGINEERING CENTER
1850 Wiehle Road
Reston, VA 22090
Attn: Manoj Dharamsi
Attn: Lt. Col. William Dlugos, R540

DEPARTMENT OF DEFENSE
9800 Savage Road
Ft. Meade, MD 20755
R. McFarland R17 (2 copies)
M. Tinto S46 (2 copies)

ELEX3101
Naval Electronics Systems Command
Department of Navy
Washington, DC 20360
Attn: Frank Deckleman

ELEX3301
Naval Electronics Systems Command
Department of the Navy
Code 3301
Washington, DC 20360
C.C. Stout
J. Machado

BOLT BERANEK AND NEWMAN INC.
1701 North Fort Myer Drive
Arlington, VA 22209
E. Wolf

DISTRIBUTION Cont'd
[QTR 16]

BOLT BERANEK AND NEWMAN INC.
50 Moulton Street
Cambridge, MA 02138

R. Alter
A. Owen
R. Binder
R. Bressler
A. Lake
J. Robinson
A. McKenzie
F. Heart
P. Santos
R. Brooks
W. Edmond
J. Haverty
E. Killian
D. McNeill
M. Brescia
A. Nemeth
B. Woznick
R. Thomas
R. Koolish
W. Milliken
S. Groff
M. Hoffman
R. Rettberg
W. Mann
P. Carvey
D. Hunt
A. Chou
S. Chiu
L. Evenchik
D. Flood Page
J. Herman
Library